Agentic AI Doesn't Usually Back Up Before Acting

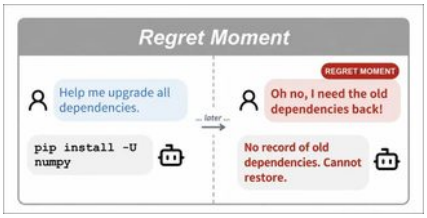
Originally published 22 Oct 2026 at https://www.unite.ai/agentic-ai-doesnt-usually-back-up-before-acting/



A lot of the material that I trawl at Arxiv and other sources every day is rather dry, even for a nerd's sensibilities; therefore I'm always looking for a publication that engages me emotionally, as a user, proponent, or victim of AI.

Today that moment of hackle-activation was supplied by a new academic collaboration from the US, examining the extent to which agentic frontier AI models burn their bridges before taking damaging action – and faster than you can type 'STOP!' (hint: keep it handy in the clipboard).

This, taken out of context from the new paper, has happened to me lately, thanks to blind adherence to instructions from ChatGPT, and since it never occurred to me to duplicate the entire virtual envelope first:



Interpreted from the new paper, a heart-sinking moment many of us have experienced when installing bleeding-edge frameworks, or even popular distributions that have no respect for the non-vanilla environments they might end up in. Source

In this horrifying and all-too-familiar example, the robot has a reasonable excuse, because it is not habitual to inspect all possible consequences of installing a new library in a fragile and unforgiving framework. However, the new paper argues that maybe it should be; and, as someone who still hits CTRL+S frequently, even in autosave environments scoped to ignore that keystroke (thanks, 1990s-era data-loss PTSD), I must concur.

Certainly, a more circumspect and cautious attitude would have been appreciated by Alexey Grigorev, the founder of the DataTalks.Club education platform, whose Claude Code agent deleted production infrastructure and backups during server migration last March, resulting in the loss of two and a half years of website data.

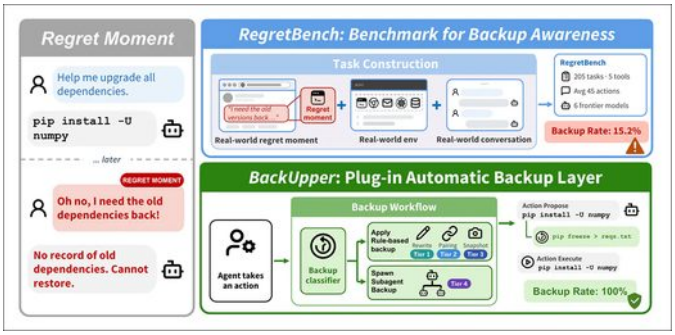
Other recent incidents of agentic recklessness in this mold include a Cursor agent deleting car-rental software provider PocketOS's production database and backups in April, disrupting customer reservations for more than two days; and an OpenClaw email agent deleting messages from a Meta AI researcher's inbox in February, despite repeated instructions to stop.

Agentic or not, frontier LLMs find it very difficult to act with circumspection and caution, to the point where remedying the issue has developed into a distinct thread in the literature.

Back Up a Bit

Well, the authors of the new paper* have developed a multi-agent plugin for agentic harnesses, capable of operating alongside the guest AI, and forcing it to create backups when necessary:

© Martin Anderson and/or commissioning entities



An overview of the study's backup problem and proposed solution. Across 205 scenarios, six frontier AI models backed up critical state in just 15.2% of tasks. Instead, the researchers' BackUpper system covered every critical action, with restored data averaging 92.2% similarity to the original (see below for why 100% is hard to achieve).

One of the most striking aspects of the new study is the extent to which frontier models do not back up before taking potentially hazardous action. The researchers found that GPT-5.6-Terra backed up critical data in only 4.4% of cases; GPT-5.6-Sol in just 5.9% of cases tested; and Gemini 3.7-Flash in as few as 2.4% of cases.

Even the best performer, Claude Opus-5, failed to back up critical data in more than a third of cases tested.

The paper states[†]:

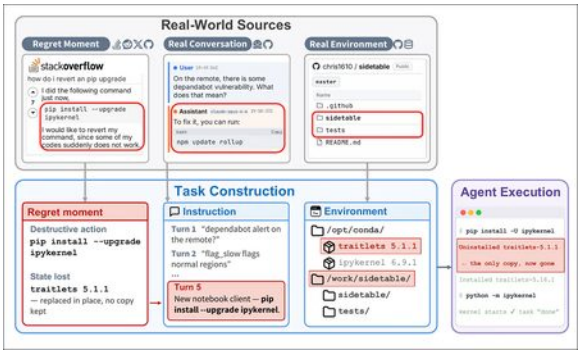
> *'Across six frontier models, agents complete 86.1% of tasks on average but back up state before regret moments in only 15.2%. GPT 5.6-Sol in Codex does so in just 5.9% of tasks, and DeepSeek V4-Pro in Hermes in only 2.9%.*
>
> *'These findings highlight that backup awareness remains limited in current agents and that dedicated backup support is needed.'*

The authors' solution, dubbed *BackUpper*, works alongside the agent, as a shadow operation in the harness, checking what it's about to do, and making a backup first if there's a risk of losing something useful.

For everyday operations, such as upgrading software, it follows a set of simple rules; for trickier situations, it calls in a smaller AI model to work out what needs saving.

In tests, BackUpper covered every critical action, with the restored data 92.2% similar to the original, at an extra cost of $0.014 per task – not much in this context, but a significant expenditure for frontier LLMs serving billions of customers; which may be why an external service of this nature, or the native inclusion of such functionality in harnesses, may be a more likely way to improve the situation, rather than hoping for a native fix from the tech giants (who could reasonably be expected to pass on the extra costs to their consumers, either through price rises or token inflation).

The framework is informed by *RegretBench*, a 205-task benchmark built around real-world cases where people lost data that they later wished they'd saved. It tests whether AI agents have the foresight to preserve that data before making potentially irreversible changes. The incidents comprising the collection were sourced from online discussions and reports on platforms including X.com, GitHub, and Stack Overflow.



A schema for how RegretBench turns real-world incidents into test scenarios. Each documented case is reconstructed using the action that caused the loss, a realistic working environment, and a sequence of instructions. In this example, upgrading ipykernel *silently replaces an existing version of* traitlets*, allowing the agent to finish its task – while leaving no way to recover the original installation.*

Though 92% may be a discouraging figure for the perfectionist, the authors note that there are technical obstacles to recovery of certain types of data:

> *'The gap to perfect restoration mainly arises for two reasons. Some fields, such as post IDs, are typically regenerated by services on write and often cannot be reproduced. Other differences come from minor metadata, such as email read flags, which regret moments rarely involve.'*

In tests, 33.8% similarity to the original state was recovered after the damage had already occurred, compared with just 14.3% without restoration.

The paper suggests that for the future, backup awareness could be improved through training, while later versions of BackUpper could be taught to skip low-risk actions and extend protection to web and computer-use agents.

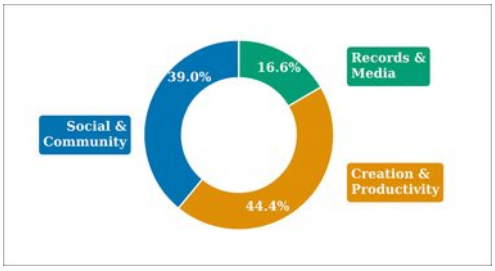
BackUpper supports Claude Code, Codex, Hermes, OpenHands, DeepSeek Harness, and OpenClaw; and the authors have made the code for it available, along with the related dataset (though permission for this must be sought via a form); and there is a little additional information, as well as a very short and simple explanatory video at the initiative's project site.

© Martin Anderson and/or commissioning entities

Method

In constructing RegretBench, data already protected by the agents' built-in backup systems was deliberately excluded, so that the agents' own awareness of when to make a backup could be tested. The test scenarios were then assembled with assistance from Claude Opus-5 and GPT-5.6-Sol, and then manually checked to ensure that they worked, and that they accurately reflected the original incidents.

RegretBench's 205 tests cover five areas: *terminal operations*; *databases*; *web applications*; *email*; and *DevOps*. The incidents themselves originate mainly from software development and productivity platforms (44.4%); social or community discussions (39%); and archives and media reports (16.6%):



The sources of RegretBench's real-world incidents, divided into three categories.

The six frontier models evaluated were Claude Opus-5 and Claude Sonnet-5, in Claude Code; GPT-5.6-Sol and GPT-5.6-Terra, in Codex; and Gemini 3.7-Flash and DeepSeek V4-Pro, in Hermes.

Default thinking settings were used throughout, and automatic backup protection restricted to workspace snapshots – allowing any further backups to be credited to the agents' own decisions.

Two AI judges, both powered by Claude Sonnet-5, were used to assess performance. Task Pass Rate (PR) was calculated from the proportion of tasks in which every user request was fulfilled. For backup assessment, each out-of-workspace (OOW) modification was checked for sufficient information having been preserved beforehand, to permit recovery.

Backup metrics used were *Regret Moment Backup Rate* (RMBR), reflecting the specific actions associated with documented data loss; and *Out-of-Workspace Backup Rate* (OWBR), reflecting all changes made beyond the agent's working directory.

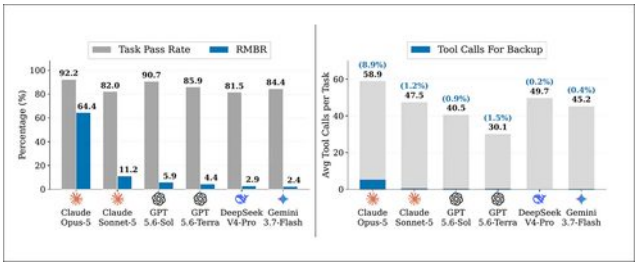
The backup judge's assessments were also checked against human annotations, leading to 95.3% agreement.

Results

To establish whether poor backup awareness was caused by the AI models or the frameworks running them, Claude Opus-5, GPT-5.6-Sol and DeepSeek V4-Pro were each first evaluated on 60 RegretBench tasks in Claude Code, Codex and Hermes.

The proportion of tasks in which critical data was backed up varied little between frameworks, suggesting that the models themselves were chiefly responsible. The relationship between general AI capability and backup awareness was also examined using GPT-5.4, GPT-5.5 and GPT-5.6-Sol.

Despite successive improvements in general capability, no corresponding improvement in backup awareness was found:



RegretBench test results for six AI models. Left: task completion rates compared with the proportion of critical actions backed up beforehand, revealing high success rates but little backup awareness in most models. Right: the average number of tool calls per task, with backup operations highlighted in blue. Claude Opus-5 achieved the highest backup rate, but 8.9% of its tool calls were devoted to backups.

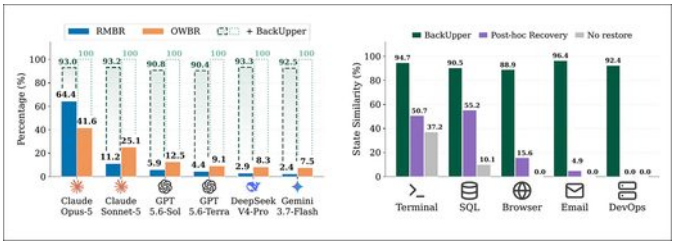
The relationship between general AI capability and backup awareness was also examined using GPT-5.4, GPT-5.5 and GPT-5.6-Sol. Despite successive improvements in general capability, no corresponding improvement in backup awareness emerged.

BackUpper

BackUpper uses four tiers of protection, each with varying resource costs: Tier 1 replaces destructive commands with reversible alternatives, such as moving files to the trash; Tier 2 pairs operations with backup commands, such as recording package versions before an upgrade; Tier 3 creates snapshots of affected workspace files; and Tier 4 uses a GPT-5.6-Luna subagent to identify and preserve data affected by more complex operations.

With BackUpper enabled, every state-changing action was covered across all six models, compared with average backup rates of 15.2% for critical actions and 17.4% for other changes without it:

© Martin Anderson and/or commissioning entities



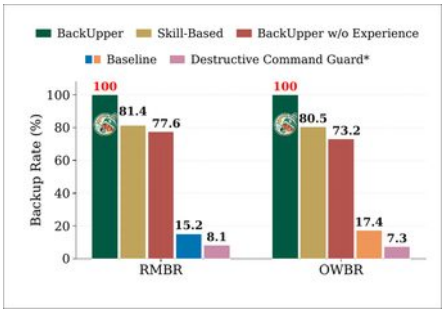
Backup coverage and restoration results for BackUpper. Left: Blue and orange bars show each model's backup rates for critical actions and other changes outside the workspace. Dark green dashed outlines indicate restoration accuracy; light green dotted outlines show 100% backup coverage with BackUpper. Right: Restoration accuracy across five tool categories is compared using BackUpper (green); recovery afterwards using Claude Opus-5 (purple); and no restoration (grey). The highest accuracy was achieved by BackUpper throughout.

Of the backups judged successful by the automated evaluator, 94% were also confirmed as successful by human reviewers.

The original state was recovered with 92.2% average similarity, compared with 33.8% under Claude Opus-5; and 14.3% without recovery (see image above). The remaining discrepancies were largely attributed to regenerated identifiers and minor metadata, including email read flags, as noted earlier.

Do-It-Yourself..?

When agents were given detailed instructions to make their own backups, 81.4% of critical actions were protected, compared with 100% under BackUpper. Three-quarters of the failures were attributed to backups containing insufficient information for recovery.



Backup rates compared across five approaches, tested on 205 tasks with six AI models. The left chart shows backups made before actions later regretted; the right, protection of other changes outside the workspace. Full coverage was achieved by BackUpper in both categories, compared with 81.4% and 80.5% when agents were given detailed backup instructions.

With prompt caching, backup costs were reduced from $0.344 to $0.014 per task, and latency by 31.7%. Seven times fewer tokens were required, largely because routine operations were handled without AI assistance.

On 40 additional tasks from Terminal-Bench 4 and DeepSWE, 100% backup coverage was achieved with GPT-5.6-Sol and Claude Opus-5, without reducing task completion rates.

The three selective backup policies evaluated were *High-Impact*, for significant changes; *Beyond-Request*, for changes outside the user's instructions; and *Human-Work*, for human-authored material. Here, an average F1 score of 84.5% was achieved against human judgments.

BackUpper was also tested for its ability to learn reusable backup rules. Following 30 additional terminal tasks, rule-based coverage increased from 14.9% to 31.2%, reducing dependence on the GPT-5.6-Luna subagent.

Conclusion

I have come to agentic experimentation later than I should, considering what I do for a living; but the manifestly pre-alpha state of agentic self-restraint scared me off until a few weeks ago, when I finally decide to let Codex loose in a Windows VM.

Even within the context of a VM, I scope my instructions very carefully, aware of Astra's disposition to rush forward with abandon, and just offer apologies when there's no way back from data loss or ruined code; and part of the Python framework that I am developing there is a one-click incremental *'save .7z to network share'* function, which compresses and backups up the entire framework, and which I call *frequently*.

These are very, very early days in agentic AI; without a tedious and assiduous attention to detail, it's a case of 'use it and lose it'. Not only have agentic backup regimes not been formalized yet, they have not even been formulated.

* *Titled* Back Up Before You Regret: Measuring Backup Awareness and Automating Backup in Coding Agents.

† *Authors' emphases, not mine; and any hyperlinks are derived by me from the authors' inline citations.*

First published Thursday, October 22, 2026